

Een belangrijk onderdeel binnen het OO programmeren zijn de object functies. Deze functies bepalen het gedrag dat een object vertoont en geven de mogelijkheid om objecten met elkaar te laten samenwerken. Via deze functies kunnen objecten met elkaar informatie uitwisselen en elkaars status (waarden van de properties) aanpassen. Verder bieden functies ook nog een ander voordeel: code die je mogelijk meerdere keren in de loop van je programma wil uitvoeren zijn gegroepeerd en op te roepen via de functienaam.

Oefening 1

Maak het spel zeeslag waarin het de bedoeling is om de verborgen boot zo snel mogelijk te vinden. Hieronder volgt eerst een beschrijving van de mogelijke klassen die je kan gebruiken en hun eigenschappen. Daarna volgt een uitleg over het verloop van het spel. De volgorde van de klassen is niet toevallig. Probeer deze volgorde ook te volgen bij het oplossen van deze oefening.

Het spel bestaat uit een zeeslag spel klasse die op zijn beurt een boot klasse beheert die een bepaalde locatie heeft binnen het spel. We kunnen met andere woorden 3 klassen onderscheiden:

1. Locatie

- Eigenschappen
 - X: x coördinaat van de locatie (int)
 - Y: y coördinaat van de locatie (int)

2. Boot

- Eigenschappen
 - Locatie: de locatie van de boot (Locatie). Een boot heeft geen lengte. De volledig boot bevindt zich op deze x y locatie.
 - Gezonken (bool)
 - False: de boot werd nog niet gezonken
 - True: de boot werd gezonken
 - Waarde: de waarde van de boot (int) (gebruik de volgende code om de waarde te bepalen: `new Random(Guid.NewGuid().GetHashCode()).Next(1, 6)`)
- Constructoren
 - Default constructor: enkel nodig wanneer de properties niet rechtstreeks worden toegewezen (vb: `public int Test { get; set; } = 0;`)
 - Gezonken = false
 - Waarde = 0
 - Locatie = nieuwe Locatie

3. Zeeslag

- Eigenschappen
 - Boot1 / Boot2 / Boot3: 3 properties van het type Boot
 - MatrixLengte: de hoogte en breedte van de zeeslag grid (int). Deze waarde mag je hard coded bepalen in je constructor of rechtstreeks toewijzen (vb 4)
 - AantalPogingen: aantal resterende pogingen van de gebruiker (int). Ook deze waarde mag je hard coded in je programma plaatsen (vb: 10)
 - Afgelopen: geeft aan dat het spel afgelopen is (bool). Wanneer is het spel afgelopen:
 - Als alle boten zijn gevonden (Gezonken status is true)
 - Als alle pogingen opgebruikt zijn (AantalPogingen = 0)
 - AantalPunten: het totaal aantal punten gewonnen door de speler (int) = de waarde van de gezonken boten
 - BotenResterend: geeft het aantal niet gezonken boten terug (int)

Al deze bovenstaande eigenschappen kunnen eventueel uitgelezen worden door een andere klasse, maar niet bepaald worden.

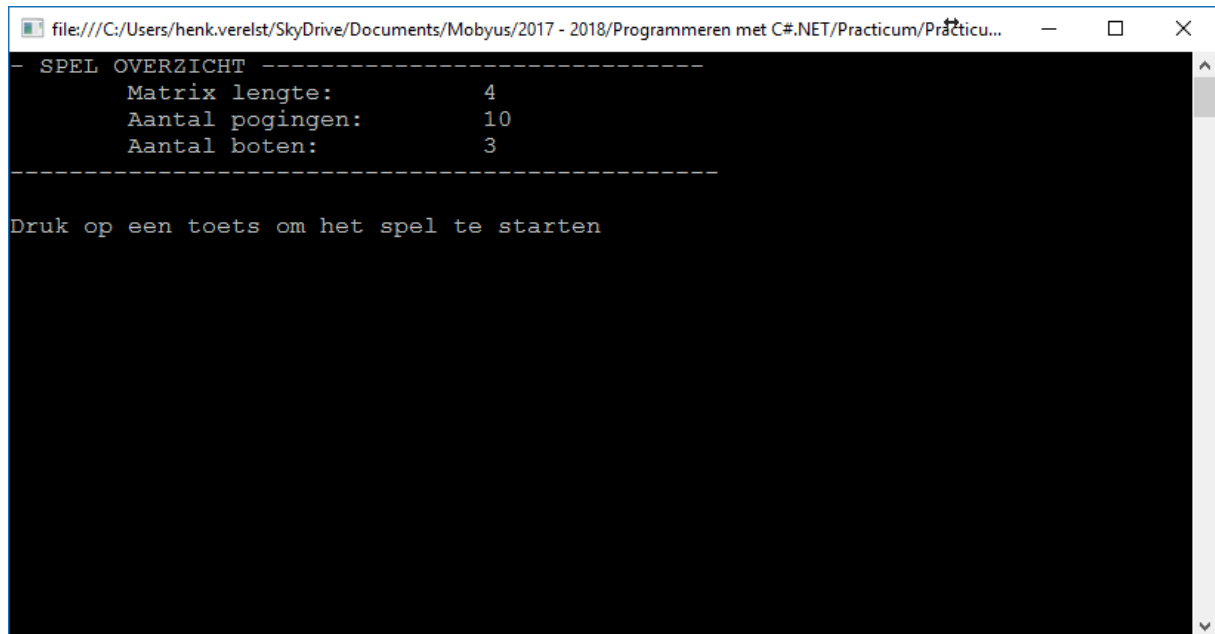
- Constructoren
 - Default constructor:
 - Initialisatie van:
 - AantalPogingen (vb 10)
 - MatrixLengte (vb 4)
 - Boot properties + instellen van locatie objecte (X en Y properties) met random object: `random.Next(1, MatrixLengte + 1)`
- Functies
 - SmijtBom:
 - Return: resultaat van de poging (int)
 - 0 – het spel is afgelopen (check van Afgelopen property)
 - 1 – coördinaten zijn buiten de grid (check van MatrixLengte property)
 - 2 - raak! (Gezonken property van de boot in kwestie moet op true worden gezet)
 - 3 - mis!

In geval van 2 & 3 moet het aantal pogingen met 1 worden verminderd.
 - Parameters: de locatie waar de bom moet gesmeten worden (Locatie object)
 - Omschrijving: Deze functie zal
 - Controleren dat het spel al dan niet is afgelopen
 - De Locatie van de bom binnen de matrix valt
 - Nakijken dat de locatie van de bom op 1 van de 3 boten is gevallen
 - GeefOmschrijving
 - Return: de omschrijving van de spel parameters (string, zie screenshots)
 - Parameters: geen
 - Omschrijving: De GeefOmschrijving functie zal een string teruggeven waarin een overzicht wordt gegeven van de onderstaande gegevens. Deze functie kan door het programma gebruikt worden om een overzicht van het spel te tonen (zie screenshots).
 - Matrix lengte
 - Aantal pogingen
 - Aantal boten (= 3)
 - GeefStatus
 - Return: de status van het spel (string, zie screenshots)
 - Parameters: geen
 - Omschrijving: De GeefStatus functie zal een string teruggeven waarin de status van het spel wordt getoond. Dit zijn:
 - Aantal resterende pogingen
 - Aantal gewonnen punten
 - Aantal resterende boten

Wanneer deze klassen aangemaakt zijn kan het programma gemaakt worden. De speler kan een bom smijten zolang het spel niet afgelopen is. Hierbij geeft hij bij elke resterende poging de X en Y-coördinaten van de locatie in en krijgt hij het resultaat te zien van zijn poging. Elke keer wordt de spel status opnieuw getoond. Wanneer het spel is afgelopen wordt de spel status nog 1 maal getoond en sluit het programma af.

Hieronder de mogelijke flow van het spel:

Start van het spel

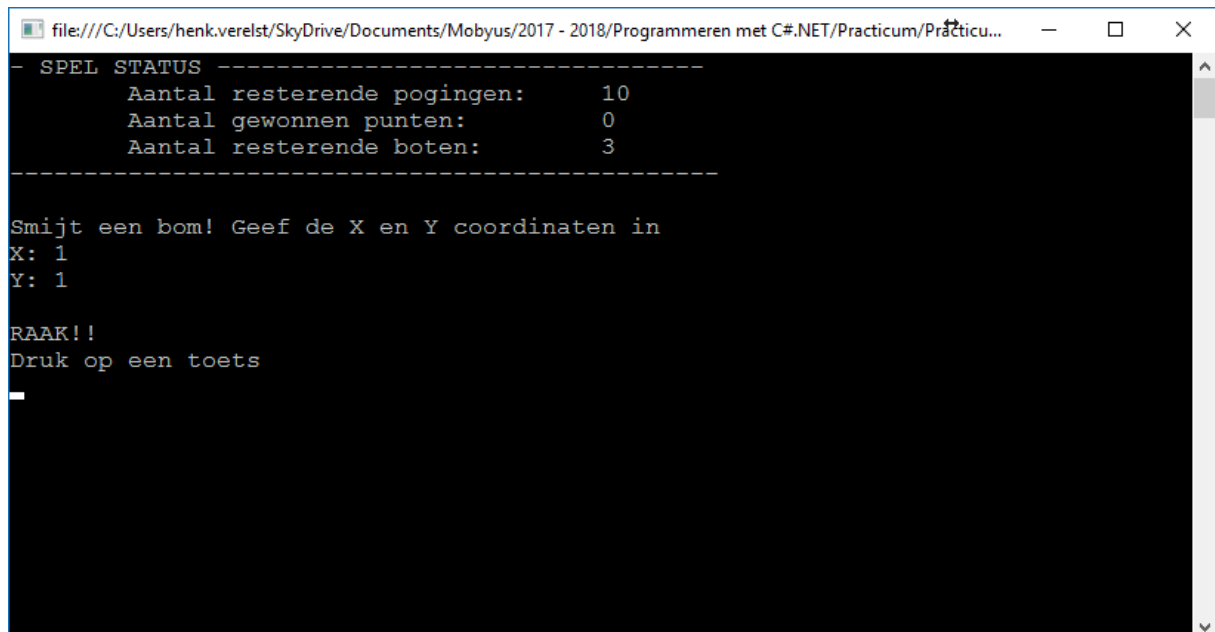


```
file:///C:/Users/henk.verelst/SkyDrive/Documents/Mobyus/2017 - 2018/Programmeren met C#.NET/Practicum/Practicu...
- SPEL OVERZICHT -----
  Matrix lengte:           4
  Aantal pogingen:        10
  Aantal boten:           3
-----

Druk op een toets om het spel te starten
```

Smijt een bom

RAAK



```
file:///C:/Users/henk.verelst/SkyDrive/Documents/Mobyus/2017 - 2018/Programmeren met C#.NET/Practicum/Practicu...
- SPEL STATUS -----
  Aantal resterende pogingen: 10
  Aantal gewonnen punten:    0
  Aantal resterende boten:   3
-----

Smijt een bom! Geef de X en Y coördinaten in
X: 1
Y: 1

RAAK!!
Druk op een toets
_
```

MIS

```
file:///C:/Users/henk.verelst/SkyDrive/Documents/Mobyus/2017 - 2018/Programmeren met C#.NET/Practicum/Practi...
- SPEL STATUS -----
  Aantal resterende pogingen:      9
  Aantal gewonnen punten:         2
  Aantal resterende boten:         2
-----

Smijt een bom! Geef de X en Y coördinaten in
X: 2
Y: 2

MIS!!
Druk op een toets
_
```

Coördinaten buiten de matrix

```
file:///C:/Users/henk.verelst/SkyDrive/Documents/Mobyus/2017 - 2018/Programmeren met C#.NET/Practicum/Practi...
- SPEL STATUS -----
  Aantal resterende pogingen:      8
  Aantal gewonnen punten:         2
  Aantal resterende boten:         2
-----

Smijt een bom! Geef de X en Y coördinaten in
X: 5
Y: 5

Verkeerde coördinaten. Maximum waarde = 4
Druk op een toets
_
```

Het spel is afgelopen

```
file:///C:/Users/henk.verelst/SkyDrive/Documents/Mobyus/2017 - 2018/Programmeren met C#.NET/Practicum/Practi...
- SPEL STATUS -----
  Aantal resterende pogingen:      0
  Aantal gewonnen punten:         0
  Aantal resterende boten:         3
- VERLOREN - Geen resterende pogingen
-----

Druk op een toets om af te sluiten
```

```
file:///C:/Users/henk.verelst/SkyDrive/Documents/Mobyus/2017 - 2018/Programmeren met C#.NET/Practicum/Practi...
- SPEL STATUS -----
  Aantal resterende pogingen:      7
  Aantal gewonnen punten:         5
  Aantal resterende boten:         0
- GEWONNEN - Alle boten werden gezonken!
-----

Druk op een toets om af te sluiten
-
```